MAY/88

ZX-Appeal

vancouver sinclair users group

next meeting:

KILLARNY COMMUNITY CENTRE
6260 KILLARNY STREET
VANCOUVER

FRIDAY:7:00PM

Friday 13th !!!
May/88

ZXAppeal is a monthly newsletter put out by the Vancouver Sinclair Users Group. For more information on the group and ZXAppeal see the backcover.

inside

Timex/Sinclair 3068.....$199

--Amiga killer available soon

Timex Disk Drive.....$49

--plugs into Timex Bus Expansion unit

.....this was the future.....almost.

See details inside.

THIS ISSUE:
.....Ken
.....Bill
.....Harvey
.....Wilf
.....Fred
and lots more!!

THIS ISSUE.....

Everyone finished scraping winter's rust off their barbeque? Started on the ol' flower beds? I know spring has sprung and summer is just up the road but don't neglect your little black or silver pal that saw you through the depths of winter.

Not alot of submissions this issue so maybe this will be the first "skinny" issue. Saying that, I'll probably come up with a bunch of neat program printouts and other stuff of interest and fatten the pot but we'll see.

We include the remainder of the WRx16 Upgrade article from last time - the BASIC and M/C listings. Fred N. drops in again, this time with his thoughts on the concept of "Shareware". Harvey T. is back to show us how we can expand our QLs in another episode of "Playing with..." Bill Rutter joins us again with Part Three of "Header Hacker". Plus a bunch of other odds and ends.

****

BITS & PIECES.....

...the Pacific Coast Computer Fair was held, very sucessfully, on Saturday, the 7th of May. VSUG was out in force with Wilf, Gerd, Rod, Louis, Eric, and Harry manning the tables. Many attendees were seen stopping to comment, with some awe and astonishment, on Sinclair machines still being around and to share the fact that the Sinclair was their first machine. We, of course, replied with the question "Why did they move on when they had it right the first time?"

...LISTing, the newsletter of the Long Island Sinclair Timex group reports some fascinating insight into the TIMEX CORP world prior to the "dumping". Apparently the President of Psion, Inc. was previously the Product Development Director for Timex. He related the following tidbits while visiting with LIST at a recent meeting: the engineering for the Bus Expansion Unit for the 2068 was COMPLETE. This was the unit that was to allow the addition of bank-switched memory, parallel and serial ports, and disk drives; the drives were to be 3.5" and were to come out of the box at $49.95!; the next machine off the line was to compete with the Amiga and was to be known as the 3068 - with 1 meg of RAM, virtual memory, and 256 hi-res colours and all for a target pricee of $199.95!; the computer line was dumped because the cash-drain caused by both R&D costs and the price-wars came at a critical time for Timex - their watch lines needed upgradeing to catch up with the rest of the watch industry or the bread and butter of the parent company might have been in jeopardy; Timex did feel some sense of responsibility to the User and decided to publish the 2068 manual even though they had pulled out of the market; and lastly that the TS1000 is STILL in production but is only available to industries as either completed boards or cased machines for control applications in processing. Thanks to L.I.S.T. and especially to Mr. Skyrme of Psion, Inc. for letting all of us take a peek at what might have been.....sigh!!!

****

RENEWING MEMBERS:

Ian McLean, Jay Mundy,
Glenn Read, John Sampson

NEW MEMBERS:

Ike Walker, Charleston, W VA
Robert Shade, Philadelphia, PA

IF THE EXPIRY DATE ON YOUR MAILING LABEL IS HIGH-LIGHTED BE SURE TO MAIL IN OR RENEW AT THE MEETING.

****

....next meeting!

MAY 88

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

April 8/88 Minutes

-by your humble scribe

As people entered the meeting room, they were treated to the vision of Bob Dennison (with a gleeful twinkle in his eye,(picture Robert Deniro in Brazil)) busily running telephone cable from some subterranean source into the room. This being Gerd's political honeymoon I will not dwell on how we finally had to harangue him into starting the meeting at 19:27. There were 23 intrepid souls present; and a couple came in later.

First off we had the President's report. Gerd reminded us about the PCCFA swap meeting coming up April 30/88 and we mustered a group to man the club tables. Look for a report next month. Also mentioned was the Products Profile show coming up June 8/9 at the PNE. An informal survey of how many people had 32K NVM's working showed that most had been set up with no problem. There is a new run of PCB's in the works, so if you want a 32K NVM, contact Wilf.

Rod Humphreys stood to tell us that we have approximately CAN$918.00 dollars in the old credit union. He is planning on going south at the 1K mark. It was mentioned to Rod by Harvey that he had been contacted by QL Australia and arrangements to add them to our Network mailing list were made. There was some discussion of the letter of J. Crumley (of Leavenworth fame) published in the last ZXAPPEAL. Rod spoke of the attempts to form the North American Timex Sinclair Association, NATSA. No motion was taken, but it seemed the group looks positively on this venture.

There was at this point the ritual dissection and vilification of her Majesty's Most Loyal Postal Snails. Ken Abramson piped up that his letter from China had arrived in 8 days. It seems Rod has been on the Post Office's case & has even developed what might be called a poisonal relation with the Customer Service Manager of the Pacific Region.

The hardware group had a 32K NVM trouble shooting session a Sunday ago & will meet again Sunday April 16. Harry Slot wants to make sure the record reads that he is not "a person who loves to be mysterious", but rather one who likes to make quite certain that he is correct before he speaks. He reports that not much is happening with the PC8300 the last couple of weeks as he has been tied up in other things. Harry then regaled us with tales of his trip to Mexico. Strictly no tech. He did come across an old IBM at Club Med, but it had no power supply and the retailer could not get there to 'fix' it for 8 months! Then there was a story of the fake metal detector at a customs gate. And the x-ray machine which was no danger to camera film. He had a great time, except for the airline food coming back home.

Kevin Kerney, the book librarian, has got rid of all his non-Sinclair material & so his load is considerably lightened. Bill Rutter says there is nothing much new with the 2068 library. Ian Mclean, the ZX81 librarian, was unable to make the meeting.

Ken Abramson stood to mention the BCIT open house currently underway. He also mentioned a Radiation Technology workshop coming up at the end of April. Ken then reached into a bag under the table and donned a Chairman Mao type hat with a big RED star on the front. Then the China Stories started to come forth. There are computer stores in China. An IBM-XT sells for the equivalent of CAN$11,000.00! The Chinese are pushing tourism. The hotel Ken stayed in was quite modern. It is Ken's opinion that "the sleeping giant has awakened" with a system of socialism and capitalism in parallel. On the way back he met a couple of Exchange Professionals; one of whom is a teacher as well & who is going to be in the Vancouver region. There are a few TV's around

the country but major appliances are scarce; not many phones either. He felt no hesitation among the people to talk to him on any subject. The power grid uses 220 Volts.

Harvey relayed a message he had seen to the effect that Fred Nachbaur has now got Landed Immigrant Status! A loud cheer resounded.

Now we had a room with a telephone line running in and the Telecommunication demonstrations proceeded amidst a cacaphony of private conversations. A splendid time was had by all.

****

THAT 'BLINKING' COMPUTER

by Ken Abramson

The latest research into eye blinking shows that the human brain has characteristics similar to our friendly ZX81/TS1000 computer! Dr. Joseph Stern of Columbia University has collected clinical evidence that refutes the old assumption that eye blinking is a relatively random process occurring only for the purpose of lubricating the eyeballs.

Dr. Stern's research shows that blinking is neither random, nor is it humidity dependent! Blinking can only occur when the brain's processing activities are interrupted. Blinking ceases when a person's brain is deep in concentration (intensive processing).

This scheme is somewhat analogous to the ZX81 ignoring the video display when it is running in FAST mode!!!

****

---WANTED---

Harvey Taylor would like to get his hands on a copy of the Sept/87 issue of QL World or at least acquire a copy of the final installment of the "Connexxions" series. If anyone is able to assist they can contact Harvey through the Editor.

HEADER HACKER (PART III) by Bill Rutter

A review of some of the common methods used to protect programs against piracy.

Normally, each block of a program is preceded by a border of thick Red and Cyan stripes. This is called a TONE LEADER and ensures that the code is taken into the memory at the correct time. By writing some machine code into the first loader, the system is bypassed, and we can now have LONGer, SHORTer, NARROWer, WIDer, and PULSed (or JERKY) tone leaders, which will not normally be copyable.

The TONE LEADER is followed by the Header, 17 bytes long, of the file attributes of the next block to be loaded. It is quite common to code in Headerless Files, or False Headers to stop copying.

A common method is to use an Anti-Merge Basic loader. This usually contains some sabotage POKES, which either crash the system, or give an 'Out of Memory' error. It is normally done by POKING an abnormally high value into the line length byte which precedes the last line. The program will run, but the MERGE routine crashes, because the last line appears to be too long.

The sabotage POKES, referred to above, cause a Reset if an error occurs (e.g BREAK key is pressed). The system variables area (23613/4) is normally used. Also 23659,which holds the number of lines where error messages are usually printed. If POKED with 0, the system will just 'sit' there, having no place to print messages. Also control codes can be embedded in the Basic to make the print either appear distorted, invisible or even display a message (usually a warning!) of some kind. It is also possible with machine code to raise or lower the BAUD rate (normally 1500).

Finally, what I term the hardware method is the infamous LENSLOK. At the beginning of the program a series of distorted random numbers is shown on screen and a special plastic lens in a holder is held to the screen, to decipher the numbers which are entered as the password. It can be difficult to use, and has been roundly condemned in the U.K.

This brief review may help to solve some of the problems, although there are many other methods in use. Get a Header Reader, and try some of the things I have outlined. Getting into that apparently unbreakable program can be very gratifying to a amateur 'Basic' hacker!

PLAYING WITH ELECTRICITY

-April 29/88
-by Harvey Taylor

I am starting to put together the elements of an expanded QL system & to this end, I have built several expansion cards. This month I will talk about the ROM expansion card. I will explore other hardware in future articles as I get it built.

One of the useful things about the QL initialization sequence is the way it checks the allocated ROM area of the memory map for the ROM flag. This allows the QL to initialize disc drives, procedures, ram disks ... anything which might be in ROM. I have spoken about the memory map in the past, however just as a quick reference:

QL MEMORY MAP

Address	Function
$000000	
	On Board ROM eg. JSU
$00BFFF	
$00C000	
	Plug in Expansion ROM (at back)
$00FFFF	
$010000	
	On board Hardware I/O
$01FFFF	
$020000	Base of on board 128K RAM
	Default screen 0
$028000	Start of system variables
$03FFFF	Top of on board RAM
	Normal expansion RAM
$0BFFFF	
$0C0000	Base of ROM area
	16-16K Expansion Cards
$0FFFFF	Top of ROM area

The format for a ROM to be initialized by QDOS is:

OFFSET	FUNCTION
00	$4AFB001 ROM Flag
04	Pointer.w to SBasic PROCedures & FUNCtions
06	Pointer.w to initialization routine
08	Ascii ROM id string

The code which does the checking for the ROM flag looks like this:

```
*
*  ROM initialization subroutine & caller
*
* the subroutine
004AC0  0C93 4AFB 0001  CMPI.L  #$4AFB0001,(A3)        * ROM flag present?
004AC6  6620            BNE     L$00004AE8             * If no: quit
004AC8  43EB 0008       LEA     $0008(A3),A1           * point to id string
004ACC  4EBA EF2E       JSR     L$000039FC             * print it
004AD0  302B 0004       MOVE.W  $0004(A3),D0           * get basic offset
004AD4  6708            BEQ     L$00004ADE             * if Null: skip
004AD6  43F3 0000       LEA     $00(A3,D0.W),A1        * point to Proc List
004ADA  4EBA 2328       JSR     L$00006E04             * add Proc.& Func
004ADE  302B 0006       MOVE.W  $0006(A3),D0           * get initalize offset
004AE2  6704            BEQ     L$00004AE8             * if Null: skip
004AE4  4EB3 0000       JSR     $00(A3,D0.W)           * go do init routine
004AE8  4E75            RTS
* (other stuff in here)
* the caller
004B04  267C 0000 C000  MOVEA.L #$0000C000,A3         * Look at ROM port first
004B0A  61B4            BSR     L$00004AC0             * Check that address
004B0C  267C 000C 0000  MOVEA.L #$000C0000,A3         * Look at Expansion ROM
004B12  61AC            BSR     L$00004AC0             * Check that address
004B14  D6FC 4000       ADDA.W  #$4000,A3              * Every 16K
004B18  B7FC 0010 0000  CMPA.L  #$00100000,A3          * End of ROM area yet?
004B1E  6DF2            BLT     L$00004B12             * If no: loop back
*
```

Getting on the QL bus has one tricky aspect. To quote the QL Technical Reference Guide (page 52):

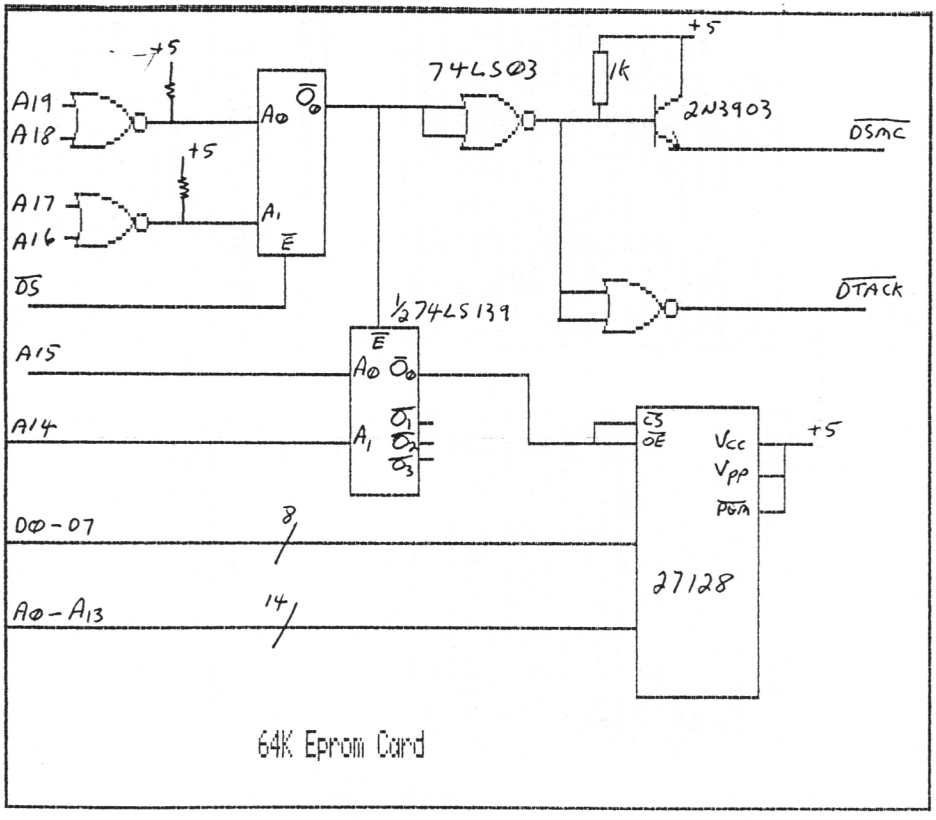
"An unexpanded QL does not look at address lines A18 and A19. In peripheral cards which are to be added to the QL, it is necessary for each card to disable the circuitry on the QL itself when that peripheral card recognizes its own address. This is achieved by pulling signal DSMCL high before DSL goes low including buffering times."

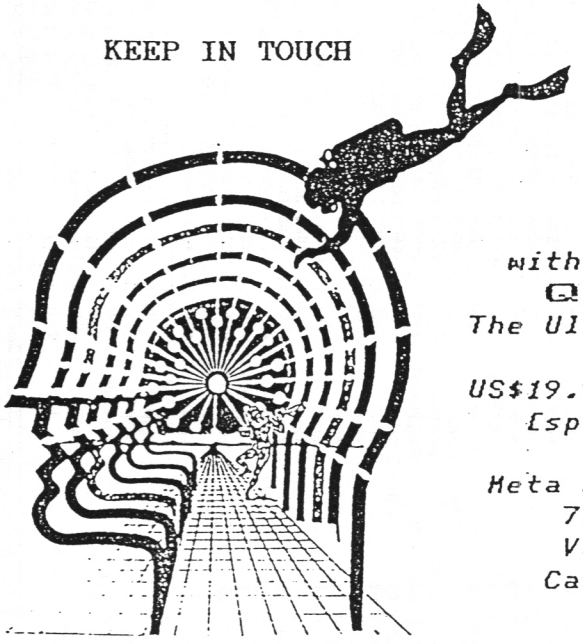
Okay that is all the background. The circuit which I have used is Figure 1. There are only 2 decoding chips used, a 74LS03 and a 74LS139. The circuit can support 4 - 16K [27128] ROMs. There is a 7805 regulator chip and a 2N3903 transistor used as well. Don't forget to use adequate bypassing.

One thing about this circuit stands out immediately. The signal DSMCL is conditioned by DSL and so cannot possibly become active before DSL! How can this be? I took the idea for this circuit from a Quanta article [August/86] and tested it. To my great surprise it worked. I have also built a Sinclair Standard type ROM interface and it works properly as well.

Note that the address of the ROM card is: $F0000 -> $FFxxx or 1111:xxAA_AAAA:AAAA_AAAA. The two bits marked "x", control the second half of the 74LS139 to decide which of up to 4 - 16K ROMs to enable. The power supply for the board was derived by using a 7805 regulator chip between the +9V rail and ground.

As usual, any questions or problems can be directed to me via the editor.





This is the conclusion to Fred and Wilf's WRx16 upgrade article from last month.

LISTING 1:
WRX16 CORE V2.1, CODE DISASSEMBLY

```
DUMMY DISPLAY FILE

addr HEXCODE  NAME MNEMONIC
==========================
4082 ED4F       DUMY LD R,A
4084 00              NOP
4085 00              NOP
4086 00              NOP
4087 00              NOP
4088 00              NOP
4089 00              NOP
408A 00              NOP
408B 00              NOP
408C 00              NOP
408D 00              NOP
408E 00              NOP
408F 00              NOP
4090 00              NOP
4091 00              NOP
4092 00              NOP
4093 00              NOP
4094 00              NOP
4095 00              NOP
4096 00              NOP
4097 00              NOP
4098 00              NOP
4099 00              NOP
409A 00              NOP
409B 00              NOP
409C 00              NOP
409D 00              NOP
409E 00              NOP
409F 00              NOP
40A0 00              NOP
40A1 00              NOP
40A2 00              NOP
40A3 00              NOP
40A4 DDE9            JP (IX)
```

```
DRIVER CODE

40A6 F3       DPLY DI
40A7 3E07          TILO LD A,07
40A9 47            TILO LD B,A
40AA 00            DP-0 NOP
40AB 10FD               DJNZ DP-0
40AD C6EF          TILO ADD A,EF
40AF 3C            DP-A INC A
40B0 20FD               JR NZ DP-A
40B2 0680               LD B,80
40B4 112000   DELY LD DE,0020
40B7 210020             LD HL,HRDF
40BA DD21C040           LD IX,DP-1
40BE 1808               JR DP-2
40C0 112000   DP-1 LD DE,0020
40C3 05              DEC B
40C4 CACF40             JP Z DP-3
40C7 19                 ADD HL,DE
40C8 7C       DP-2 LD A,H
40C9 ED47               LD I,A
40CB 7D                 LD A,L
40CC C382C0             JP C082
40CF ED52     DP-3 SBC HL,DE
40D1 ED52     DELY SBC HL,DE
40D3 23       DELY INC HL
40D4 2A0C40             LD HL,(DFIL)
40D7 11F782             LD DE,82F7
40DA 19                 ADD HL,DE
40DB 3E1E               LD A,1E
40DD ED47               LD I,A
40DF 3EF5               LD A,F5
40E1 010702             LD BC,0207
40E4 CDB502             CALL DP-5
40E7 CD9202             CALL MRGN
40EA CD2002             CALL DEND
40ED DD21A640           LD IX,DPLY
40F1 C3A402             JP POPS
40F4 DD21A640 HRES LD IX,DPLY
40F8 C9                 RET
40F9 3E1E     NRML LD A,1E
40FB ED47               LD I,A
40FD DD218102           LD IX,0281
4101 C9                 RET
```

TABLE 1: WRX16-V2 CORE (DECIMAL)

```
START=16514    BYTES=128
237:079:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:221:233:243:062:007:071:
000:016:253:198:239:060:032:253:
006:176:017:032:000:033:000:032:
221:033:192:064:024:008:017:032:
000:005:202:207:064:025:124:237:
071:125:195:130:192:237:082:237:
082:035:042:012:064:017:247:130:
025:062:030:237:071:062:245:001:
007:002:205:181:002:205:146:002:
205:032:002:221:033:166:064:195:
164:002:221:033:166:064:201:062:
030:237:071:221:033:129:002:201:
```

TABLE 2: BI-PLOT WRX16 CORE

```
START=16514    BYTES=128
237:079:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:000:000:000:000:000:000:
000:000:221:233:243:062:007:071:
000:016:253:198:242:060:032:253:
006:128:017:032:000:033:000:032:
058:052:064:230:001:023:023:023:
023:132:103:000:221:033:204:064:
024:008:017:032:000:005:202:219:
064:025:124:237:071:125:195:130:
192:237:082:237:082:035:042:012:
064:017:049:130:025:062:030:237:
071:062:245:001:007:008:205:181:
002:205:146:002:205:032:002:221:
033:166:064:195:164:002:000:000:
```

OPTIONAL SYNCHRONIZING ROUTINE
CALL THIS BEFORE CHANGING MODES
TO PREVENT SCREEN FLICKER.

```
E.G. CALL SYNC
     JP HRES
OR
     CALL SYNC
     JP NRML

addr HEXCODE  NAME MNEMONIC
==========================
4105 CD2B0F   SYNC CALL SLOW
4108 FD4634        LD B,(FRMS)
410B 3A3440   NFM? LD A,(FRMS)
410E B8            CP B
410F 28FA          JR Z NFM?
4111 C9            RET
```

TABLE 3: BIPLOT SERVICE ROUTINES

```
START=16648    BYTES=177
033:062:000:024:008:033:062:001:
024:003:033:230:001:034:189:064:
205:168:065:221:033:166:064:201:
205:168:065:221:033:129:002:201:
000:033:000:032:024:003:033:000:
048:001:000:016:054:000:011:084:
093:019:237:176:201:033:132:064:
001:032:000:126:198:128:119:024:
237:038:032:024:022:038:048:024:
018:205:073:065:024:247:038:032:
024:015:038:048:024:011:205:086:
065:024:247:253:203:059:214:024:
004:253:203:059:150:237:075:054:
064:062:128:061:144:218:173:014:
087:089:203:058:203:027:203:058:
203:027:203:058:203:027:046:000:
025:086:121:230:007:060:001:000:
008:203:002:203:017:061:032:012:
253:203:059:086:040:004:203:193:
024:002:203:129:016:235:113:201:
205:043:015:253:070:052:058:052:
064:184:040:250:062:030:237:071:
201:
```

LISTING 2: MACHINE-CODE LOADER

```
9000 REM LOADER
9010 FAST
9020 CLS
9030 PRINT "START? ";
9040 INPUT ST
9050 PRINT ST,"BYTES? ";
9060 INPUT BY
9070 PRINT BY
9080 FOR N=ST TO ST+BY-1
9090 INPUT V
9100 POKE N,V
9110 PRINT ( STR$ (V+1000))(2 TO );":";
9120 NEXT N
```

LISTING 3
BI-PLOT M.L. routine

```
40A6 F3       DPLY DI
40A7 3E07          TILO LD A,07
40A9 47            TILO LD B,A
40AA 00            DP-0 NOP
40AB 10FD               DJNZ DP-0
40AD C6F2          TILO ADD A,F2
40AF 3C            DP-A INC A
40B0 20FD               JR NZ DP-A
40B2 0680               LD B,80
40B4 112000   DELY LD DE,0020
40B7 210020             LD HL,HRD1
40BA 3A3440             LD A,(FRMS)
40BD E601          BIT0 AND 01
40BF 17                 RLA
40C0 17                 RLA
40C1 17                 RLA
40C2 17            M*16 RLA
40C3 84                 ADD A,H
40C4 67                 LD H,A
40C5 00            DELY NOP
40C6 DD21CC40           LD IX,DP-1
40CA 1808               JR DP-2
40CC 112000   DP-1 LD DE,0020
40CF 05              DEC B
40D0 CADB40             JP Z DP-3
40D3 19                 ADD HL,DE
40D4 7C       DP-2 LD A,H
40D5 ED47               LD I,A
40D7 7D                 LD A,L
40D8 C382C0             JP C082
40DB ED52     DP-3 SBC HL,DE
40DD ED52               SBC HL,DE
40DF 23                 INC HL
40E0 2A0C40             LD HL,(DFIL)
40E3 113182             LD DE,8231
40E6 19                 ADD HL,DE
40E7 3E1E               LD A,1E
40E9 ED47               LD I,A
40EB 3EF5               LD A,F5
40ED 010708             LD BC,0807
40F0 CDB502             CALL 02B5
40F3 CD9202             CALL 0292
40F6 CD2002             CALL 0220
40F9 DD21A640           LD IX,DPLY
40FD C3A402             JP POPS

4100 213E00   DPL1 LD HL,003E
4103 1808               JR STOR
4105 213E01   DPL2 LD HL,LDA1
4108 1803               JR STOR
410A 21E601   DP12 LD HL,01E6
410D 22BD40   STOR LD (BITO),HL
4110 CDA841   HRES CALL 41A8
4113 DD21A640           LD IX,DPLY
4117 C9                 RET
4118 CDA841   NRML CALL 41A8
411B DD218102           LD IX,0281
411F C9                 RET
```

```
4120 00              NOP
4121 210020   CLS1 LD HL,HRD1
4124 1803               JR CLS*
4126 210030   CLS2 LD HL,HRD2
4129 010010   CLS* LD BC,1000
412C 3600               LD (HL),00
412E 0B       FILL DEC BC
412F 54                 LD D,H
4130 5D                 LD E,L
4131 13                 INC DE
4132 EDB0               LDIR
4134 C9                 RET
4135 218440   RVRS LD HL,COL1
4138 012000             LD BC,0020
413B 7E                 LD A,(HL)
413C C680               ADD A,80
413E 77                 LD (HL),A
413F 18ED               JR FILL
4141 2620     PLT1 LD H,20
4143 1816               JR PLOT
4145 2630     PLT2 LD H,30
4147 1812               JR PLOT
4149 CD4941   PL12 CALL PLT1
414C 18F7               JR PLT2
414E 2620     UPL1 LD H,20
4150 180F               JR UNPL
4152 2630     UPL2 LD H,30
4154 1808               JR UNPL
4156 CD5641   UP12 CALL UPL1
4159 18F7               JR UPL2
415B FDCB3BD6 PLOT SET 2,(CDFG)
415F 1804               JR PLT*
4161 FDCB3B96 UNPL RES 2,(CDFG)
4165 ED4B3640 PLT* LD BC,(COOR)
4169 3E80               LD A,80
416B 3D                 DEC A
416C 90                 SUB B
416D DAADOE             JP C ERRB
4170 57       GADR LD D,A
4171 59                 LD E,C
4172 CB3A               SRL D
4174 CB1B               RR E
4176 CB3A               SRL D
4178 CB1B               RR E
417A CB3A               SRL D
417C CB1B               RR E
417E 2E00               LD L,00
4180 19                 ADD HL,DE
4181 56                 LD D,(HL)
4182 79                 LD A,C
4183 E607               AND 07
4185 3C                 INC A
4186 010008             LD BC,0800
4189 CB02     BTLP RLC D
418B CB11               RL C
418D 3D                 DEC A
418E 200C               JR NZ 41A4
4190 FDCB3B56           BIT 2,(CDFG)
4194 2804               JR Z RSBT
```

```
4196 CBC1     STBT SET 0,C
4198 1802               JR 41A4
419A CB81     RSBT RES 0,C
419C 10EB               DJNZ BTLP
419E 71                 LD (HL),C
419F C9                 RET
41A0 CD2B0F             CALL SLOW
41A3 FD4634             LD B,(FRMS)
41A6 3A3440             LD A,(FRMS)
41A9 B8                 CP B
41AA 28FA               JR Z 41A6
41AC 3E1E               LD A,1E
41AE ED47               LD I,A
41B0 C9                 RET
```

LISTING 4
BI-PLOT
BASIC

```
  5 REM WRX16-V2D DUAL HIRES
 10 LET HR1 =16648
 11 LET HR2 =16653
 12 LET HR12=16658
 13 LET HRES=16664
 14 LET NRML=16672
 15 LET CLS1=16681
 16 LET CLS2=16686
 17 LET RVRS=16701
 18 LET PLT1=16713
 19 LET PLT2=16717
 20 LET PL12=16721
 21 LET UPL1=16726
 22 LET UPL2=16730
 23 LET UP12=16734
 24 LET XP  =16438
 25 LET YP  =16439
100 REM greyplot demo
105 FAST
108 REM CLEAR BOTH DF*S
110 RAND USR CLS1
120 RAND USR CLS2
130 FOR I=10240 TO 10271
135 REM H-LINE, DF1
140 POKE I,255
145 REM H-LINE,DF2
150 POKE I+5984,255
160 NEXT I
165 REM V-LINE, BOTH DF*S
170 POKE XP,4
180 FOR I=0 TO 127
190 POKE YP,I
200 RAND USR PL12
210 NEXT I
215 REM V-MARKERS
220 FOR I=10080 TO 8192 STEP -192
230 POKE I,60
232 POKE I+2048,60
235 POKE I+4096,60
237 POKE I+6144,60
240 NEXT I
245 REM H-MARKERS
250 FOR I=4 TO 244 STEP 10
260 POKE XP,I
270 FOR J=2 TO 6
280 POKE YP,J
290 RAND USR PLT2
300 POKE YP,J+59
310 RAND USR PLT1
320 NEXT J
330 NEXT I
340 REM LO-RES AXES
350 FOR I=0 TO 63
360 PLOT I,12
370 PLOT I,27
375 IF I)12 AND I(43 THEN PLOT 0,I
380 NEXT I
390 REM PLOT CURVES
395 REM F1:SINE CURVE
400 LET A$=" SIN (H* PI /40)"
405 REM F2:DECAYING EXPONENTIAL
410 LET B$=" EXP (-H/80)"
415 FOR H=0 TO 251
420 LET Z1= VAL A$
425 LET Z2= VAL B$
430 POKE XP,H+4
435 REM FUNCTION NO. 1 (DF 1)
440 POKE YP,63+60*Z1
445 RAND USR PLT1
450 REM FUNCTION NO.2 (DF 2)
455 POKE YP,63+60*Z2
460 RAND USR PLT2
465 REM FCTN.1 * FCTN.2 (BOTH)
470 POKE YP,63+60*Z1*Z2
475 RAND USR PL12
480 REM LO-RES PLOTTING
485 PLOT INT (H/4)+1,27+15*Z1
490 PLOT INT (H/4)+1,27+15*Z2
495 PLOT INT (H/4)+1,27+15*Z1*Z2
520 NEXT H
525 POKE 16418,0
590 PRINT AT 17,0;
593 PRINT "********************************"
594 PRINT ";1; DISPLAY DF1  2; DISPLAY DF2;"
595 PRINT ";3; DISPLAY BOTH DISPLAY FILES ;"
596 PRINT ";4; REVERSE HIGH - RES DISPLAYS;"
597 PRINT ";5; RETURN TO NORMAL LR DISPLAY;"
598 PRINT ";6; QUIT TO BASIC"; TAB 31;";"
599 PRINT "********************************"
600 RAND USR HR12
610 IF INKEY$ () "" THEN GOTO 610
620 LET Z$= INKEY$
630 IF Z$="" THEN GOTO 620
640 IF Z$="1" THEN RAND USR HR1
650 IF Z$="2" THEN RAND USR HR2
660 IF Z$="3" THEN RAND USR HR12
670 IF Z$="4" THEN RAND USR RVRS
680 IF Z$="5" THEN RAND USR NRML
690 IF Z$ () "6" THEN GOTO 610
700 RAND USR NRML
710 STOP
9990 LET S$="BIPLOT-DEMO"
9991 SAVE S$
9996 PRINT "PLEASE WAIT ABOUT 75 SECONDS...."
9997 PAUSE 200
9998 CLS
9999 RUN
```

SHAREWARE — CAN IT WORK?

(Or, "How To Get Tons of Software for Free... Almost.")

by F. Nachbaur

Software has always been a difficult commodity to market and distribute. Users want new software, they want support for their software purchases. What's more, EVERY computer buff likes to build up a big library of programs. Most are never or rarely used, but there is a certain element of the pack rat in all of us.

Suppliers for obsolete computers spend time and effort to produce new products for a dwindling customer base. It has been proven again and again that one stands a slim chance of making a profit in these "low-end" computers. Still, some will try. As long as there are buyers, there will be sellers. However, dealers and programmers want their efforts to be financially worthwhile.

These ideas have often clashed, with the one camp copying software for friends because the program is too expensive, and the other claiming that software was so expensive because many sales are "lost" to illegal clones. This affects not only our tiny corner of the computer world, but all computers and computerists. A resolution could come in a new software marketing concept called "Shareware" or "Fairware."

Copying programs is a fact of life. You do it. I do it. I've only actually paid for a portion of the programs I have in my collection. It is important to note, however, that out of these programs are a scant half-dozen I use all the time, and maybe a couple dozen I use occasionally. The rest were looked at once or twice, before being enshrined in "the files" (one of two big drawers full of tapes).

The fundamendal tenet is, "If you use it, pay for it." The most direct application of the "shareware principle" is to send a "shareware contribution" to the author of a cloned program that you are finding useful. (Obviously, no-one expects you to pay for something that you can't or don't want to use.) Unfortunately, it can be a real chore to locate the actual author of a given program, so all too often we either can't find a program at all, or if we do, we may not know where to send "whatever it's worth." Furthermore, human nature being what it is, it's all too easy to indefinitely procrastinate mailing that contribution.

When we freely clone programs from our friends, we have no recourse if we are missing part of the documentation, or if there are updates or improved versions available. How do you get unanswered questions answered, bugs debugged or at least de-fanged, gain added technical or background information? How do you reward the creators and developers of the programs you have in your library? Some programs are the product of several diverse individuals; how do you reward each one's part in the product? At the same time, how do you keep your computer from driving you into the poor-house?

The answer to these questions might be to expand and organize the concept of "shareware." I don't know if formal guidelines have ever been published, but "shareware" or "fairware" libraries are simply an organized way of paying fairly for what you are already using, and to make other programs available to you at reasonable cost. Shareware libraries are already in operation all over the continent.

The libraries freely exchange material, or even share a database of available programs. You purchase from your closest library, at a reasonable cost per program. In some cases, users can submit written updates, clarifications, and other discoveries to be included in the documentation file for any given program.

Dubbing, mailing and otherwise processing orders also needs to be covered and made worthwhile. The library therefore takes a moderate fee for these services, and splits the rest with the authors who contributed.

This is often done quarterly, with payments made in the quarter following the order-taking. A sliding scale can determines how much each author receives; obviously, a massive application should pay better than a simple arcade game. In principle, then, everyone gets a fair share of whatever interest remains in the field of users.

THE NEW ZX WORLD

There is something oddly "different" about the ZX81 family of computers, which spawned a massive wave of interest in the early '80's. Partly because the machine was improperly marketed, partly because development of the TS2068 was completed too late, and largely because of the overall market slump, Timex closed down the Timex Computer division. The TS2068 was in reality never truly completed, and new products that were prototyped never saw the light of day.

Yet in Europe, the ZX Spectrum, the prototype of the TS2068, was the most popular personal computer. (Never mind that it was mostly used for games, and never saw true development of its potentials.) So now we have two quite different machines, one very simple yet powerful, and the other even more powerful, both sharing that unique invention called Sinclair ZX BASIC. Both are proving to retain a lot of interest in a hard core of dedicated fanciers.

What's more, there are still new-comers to Sinclair-Timex computing. The TS1000 surplus is still being doled out into unsuspecting homes, and 2068's are around if you look for them. Other machines are "hand-me-downs" or are picked up at garage sales. Some of these machines will strike that same chord in a brand-new user, that it struck in all of us when we first started playing with them. There's something very neat about these little wonders.

Our ace-in-the-hole is that the ZX81 is simple enough to build completely from the ground up, and could be built from completely stock parts. By looking back at the ZX80 schematic, you realize that this machine will never become a fossil because of blown custom ULA or SCLD chips. (Wags might say it's because it's always BEEN a fossil.)

At any rate, we have a lot going for us. I think that a workable shareware system can be built up. It would involve a lot of work for many members of our community, but would be worth doing for the long-term rewards.

There are a number of ways of implementing this, but here are some comments that just might help to make it work for our little world.

HERE'S THE PLAN

The first step is to work up a common data-base of existing programs, cross-referenced and linked to author. In the case of multiple authorship, i.e. "chain" programs, the libraries are allowed a certain discretion in assigning merit for different levels of contribution.

Another massive chore will be to find the authors. This could be tricky in some instances, since many of these guys have given up their Timexes for a Mac at home and IBM's at work. Many have all but forgotten their experience with the ZX81 and Spectrum machines, perhaps because they'd rather not remember how hard they worked and how little they got out of it.

We have to find these people, contact them, and interest them in the shareware notion. Most would be happy to release their "old" material, (it's a cheap way of gaining immortality!), especially if it means a token payment four times a year.

Some of them will regain interest, and will be inspired by the existence of the network into writing new programs. If the new programs are a hit, well then their royalty check will be appropriately bigger over the next few quarters.

Permission to place programs into shareware should be solicited on behalf of the entire library network. Similarly, libraries should adhere to an agreed-upon code of ethics regarding pricing, royalty payments, interchange and competition with other libraries.

We have to agree upon the billing formula. How do we keep it fair, yet reasonably simple? How much are users willing to pay, and how little are authors and libraries willing to receive? As a side-note, there could easily be instances of programs costing MORE through shareware, than they did when being retailed in the past.

Some of the other problems we'll run into include topics like media; obviously a tape dubbing operation is more cost-intensive than if programs are supplied on disk. Note that since the software library gets a percentage based on dubbing and photocopying time, the more efficient its information transferring method is, the better off we'll all be. TS2068 libraries would be at a distinct advantage because of shorter saving/loading times. Perhaps we should adopt a standardized fast-load tape routine for both computer types. How about making TS1000 programs available in TS2068 format? (See Kent Cook's article in SWN 5:1.) Customers buying in this format would get a price-break.

A shareware library, to be viable, should have all programs rapidly accessible on disk. Which disk standards are supported will depend on which system that each library has available. For instance, at this writing I could supply tapes or Compusa disks for ZX81, with Larken disks being the next stage. I could therefore be a librarian for this subset of Timex users.

How do we most effectively use the modem options? Can some if not all of our shareware transactions be conducted via FidoNet or Compuserve?

How do we catalog the entries, and their various versions? How do we agree on a catalog system? Or do we need to?

(As an example of this last point, I have recently decided to catalog two drawers of tapes and a bunch of disordered disks. At this point, my VU-FILE on archived programs has over 400 entries. I estimate that a total of 600 entries will result from just the stuff that I have. [Some of it was even written by me.] Many have multiple versions, of course; the record-holder is Memotext, with some 30 different versions, not counting the help files).

Assuming that we can cross these hurdles, the overall snowballing effect might just surprise us all. We have discussed this briefly at SWN, and were excited with the possibiliies. Imagine... all or almost all programs ever written available to anyone, with their creators getting a fair slice of the pie.

I'm sure that I speak for everyone at SyncWare News, when I say that we will definitely be involved in some way, perhaps as a shareware library co-ordinator. We will try to set up proposed guidelines for shareware library facilities. I'm prepared to be a "node," with my collection of ZX programs, and I'm sure that Jeff, Basil and Tom would be happy to become personally involved at some level also.

You asked us to move you, shake you. We're moving, are you shaking yet? If you felt even the slightest tremor, let us know. This just might be what it takes to allow our machines to survive, and open the door to new machines that have even more capability.

NNN ZX-FILES UPDATE

The following shareware files are available on the NNN, (604)354-4666. Use the (L)isting function from the (F)iles menu for a brief description. Note, many are HI-RES, and many have never been seen before.

Filename	size	date	count
ZXABC123.PGM	12K	2-01-88	27663
ZXAKMAN.PGM	9K	2-01-88	27662
ZXBATTLE.PGM	7K	12-17-87	24880
ZXBIGRLE.PGM	1K	2-24-88	18839
ZXBIGRLE.VAR	2K	2-24-88	18835
ZXBIPLOT.PGM	4K	2-03-88	19848
ZXBLOCK.DOC	2K	12-17-87	24893
ZXBLOCK1.PGM	2K	12-17-87	24889
ZXBLOCK2.PGM	10K	12-17-87	24845
ZXBUGBUR.PGM	7K	12-31-87	25041
ZXCOPY.PGM	2K	1-05-88	23033
ZXDANPGM.DOC	5K	12-17-87	24883
ZXDANROC.PGM	11K	12-17-87	24838
ZXDANRVG.PGM	11K	12-17-87	24836
ZXHRMATH.PGM	16K	1-18-88	19019
ZXHRWAVE.PGM	16K	1-19-88	18969
ZXHRWAVE.VAR	5K	1-19-88	18970
ZXINVADR.PGM	10K	12-17-87	24843
ZXMOUSE.DOC	4K	12-17-87	24888
ZXMOUSE.PGM	1K	12-17-87	2490
ZXMOUSE.VAR	8K	12-17-87	24857
ZXMRDATA.PGM	11K	3-12-88	3392
ZXPATCH.DOC	7K	1-09-88	21670
ZXPATCH.PGM	1K	1-09-88	21673
ZXSAVRLE.PGM	1K	2-24-88	18835
ZXSCR.PGM	8K	1-12-88	19690
ZXTMTXT.DOC	15K	12-17-87	24820
ZXTREK.PGM	8K	1-12-88	19677
ZXTRUCK.PGM	10K	12-17-87	24844
ZXWARN.PGM	3K	2-22-88	20146
ZXXON.PGM	15K	2-01-88	27661

RLE PICTURE FILES

DENEUVE.RLE	7K	12-17-87	24883
DINOSAUR.RLE	12K	2-26-88	18606
FRACTAL1.RLE	8K	12-17-87	24845
FRACTAL2.RLE	25K	2-24-88	18835
GARFIELD.RLE	11K	12-17-87	24843
GRIFFIN.RLE	4K	12-17-87	24884
MAGIC.RLE	14K	12-17-87	24823
MIDDLE.RLE	9K	2-25-88	21402
SPOCK.RLE	13K	2-25-88	21400
STARTREK.RLE	8K	12-17-87	24852
STREK1.RLE	7K	2-25-88	21405
STREK2.RLE	7K	2-25-88	21406
STREK3.RLE	6K	2-25-88	21409
STREK4.RLE	9K	2-25-88	21402
STREK5.RLE	5K	2-26-88	18612
STREK6.RLE	9K	2-26-88	18609
STREK7.RLE	11K	3-01-88	18786
STREK8.RLE	4K	3-01-88	18788
STREK9.RLE	4K	3-01-88	18789

A SPECIAL PRESS RELEASE

FROM: CCAT/S USER GROUP
OREGON CITY, OR 97045
AND

FOR RELEASE: July 1, 1988

LOCAL COMPUTER BUFFS TO ATTEND COMPUTER FAIR TO BE HELD AT THE COSMOPOLITAN HOTEL IN PORTLAND, OREGON AUGUST 6TH AND 7TH 1988.

Joint sponsors, RMG Enterprises of Oregon City, and Time Designs Magazine of Colton, Oregon, announced the Third Annual International/Great Northwest TS Mini-Fair to be held the weekend of August 6th and 7th 1988 at the Cosmopolitan Hotel in Portland, Oregon.

The event will include door prizes, vendor and user group booths, seminars on specialty programming, hardware tips and uses of all Timex and Sinclair computers, a round-table discussion, tours of the local scenic area and the famous Portland Zoo.

A number of nationally known vendors will be exhibiting at the fair, including such notables as RMG ENTERPRISE, TIME DESIGNS MAGAZINE, AMERICAN MICRO CONNECTION AND GREY & CLIFFORD COMPUTER PRODUCTS among others.

The seminars will cover such topics as Machine Code programming the Z80 microprocessor chip, Architecture of the 68000 CPU, using Archive database to its FULL capacity, Telecommunications, GIF graphics, an overview of the QL-What it is-What it could be,CP/M on TS computers, and others. Seminar speakers include Mike de Sosa, author of TAKING THE QUANTUM LEAP and QL advocate, Syd Wyncoop, author of the Z80 series of articles in TIME DESIGNS MAGAZINE as well as S & K Software titles including THE KRUNCHER, TRACER and EXPRESS. Michael Carver, current president of CCAT/S, a Timex/Sinclair User Group, Vincent Lyon, author of ARCHIVE MASTER, and several programms for the Timex 2068 and Sinclair QL. And let's not forget Ed Grey of Grey & Clifford, one of our telecommunications supporters, or Jack Dohany, of 2068 FairWare fame.

Scheduled tours for attendees and their families include a Saturday tour of the Columbia River Gorge and a Sunday tour of Portland's famous Washington Park including the Portland Zoo, the World Forestry Center, Oregon Museum Of Science And Industry, the Washington Park Rose and Japanese Gardens. These tours will be provided at a VERY nominal cost.

Local attendees will include...(enter your info here).....

Additional information is available from RMG Enterprises, Time Designs Magazine or the Cosmopolitan Hotel.

```
   1 REM program is off Clevelan
d group tape. No author is
acknowledged
   5 REM "CRYPTO"
  10 FOR i=65528 TO 65534
  12 POKE i,0
  14 NEXT i
  16 CLS : POKE 23609,30: POKE 6
5535,128: POKE 23658,8
  20 PRINT AT 9,8;"""CRYPTO"" PA
D": PRINT : PRINT : PRINT : PRIN
T
  22 PRINT "  A SIMPLE-SUBSTITUT
ION Cipher              Scratchpad"
: PRINT : PRINT : PRINT
  24 PRINT "  Enter Message or C
ryptogram-"
  30 LET c$="ABCDEFGHIJKLMNOPQRS
TUVWXYZ"
  50 INPUT LINE a$
  55 LET a$=a$+" "
  60 CLS
  70 PRINT
 100 FOR i=33 TO 1 STEP -1
 110 IF i>LEN a$ THEN LET i=LEN
a$
 120 IF a$(i)=" " THEN GO TO 200
 130 NEXT i
 150 LET r$=a$( TO 32): LET a$=a
$(33 TO ): GO TO 300
 200 LET r$=a$( TO i-1)
 210 LET a$=a$(i+1 TO )
 300 PRINT r$
 320 PRINT : PRINT
 325 IF LEN a$=0 THEN GO TO 360
 330 IF a$(1)<>" " THEN GO TO 10
0
 340 LET a$=a$(2 TO )
 350 GO TO 330
 360 PRINT AT 21,0; FLASH 1;"SET
TING PUNCTUATION"
 365 FOR r=1 TO 20 STEP 3
 370 FOR c=0 TO 31
 375 LET p=CODE SCREEN$ (r,c)
 380 IF p=0 THEN GO TO 480
 385 IF p=32 THEN GO TO 395
 390 IF (p<65 OR p>90) THEN PRIN
T AT r+1,c;SCREEN$ (r,c)
 395 NEXT c
 396 NEXT r
 480 PRINT AT 21,0;"LET "; FLASH
 1;"?"; FLASH 0;"=";c$
 490 LET h=0
 495 BEEP .03,20
 500 IF INKEY$<>"" THEN GO TO 50
0
 501 IF INKEY$="" THEN GO TO 501
 502 LET l$=INKEY$
 504 IF l$>"£" OR l$<" " THEN GO
 TO 500
 505 BEEP .02,15
 510 PRINT AT 21,4;l$
 515 FOR r=1 TO 20 STEP 3
 520 FOR c=0 TO 31
 530 IF SCREEN$ (r,c)=l$ THEN PR
INT AT r,c; FLASH 1;l$: BEEP .03
,30: LET h=1
 540 IF CODE SCREEN$ (r,c)=0 THE
N GO TO 595
 550 NEXT c
 590 NEXT r
 595 BEEP .01,10
 600 IF NOT h THEN GO TO 480
 605 INPUT "Substitute which let
ter?";m$
 610 IF m$="" THEN GO TO 605
 612 LET m$=m$(1)
 613 IF m$<" " OR m$>"£" THEN GO
 TO 605
 615 IF NOT (m$>"@" AND CODE m$<
91) THEN GO TO 650
 618 IF c$(CODE m$-64)=" " THEN
GO TO 605
 630 LET c$(CODE m$-64)=" "
 650 FOR r=1 TO 20 STEP 3
 655 FOR c=0 TO 31
 660 IF SCREEN$ (r,c)<>l$ THEN G
O TO 670
 662 PRINT AT r,c; FLASH 0;l$
 664 LET n$=SCREEN$ (r+1,c)
 666 IF (n$>"@" AND CODE n$<91)
THEN LET c$(CODE n$-64)=n$
 668 PRINT AT r+1,c;m$: BEEP .01
,27
 670 IF CODE SCREEN$ (r,c)=0 THE
N GO TO 480
 680 NEXT c
 690 NEXT r
9000 SAVE "CRYPTO" LINE 10
```

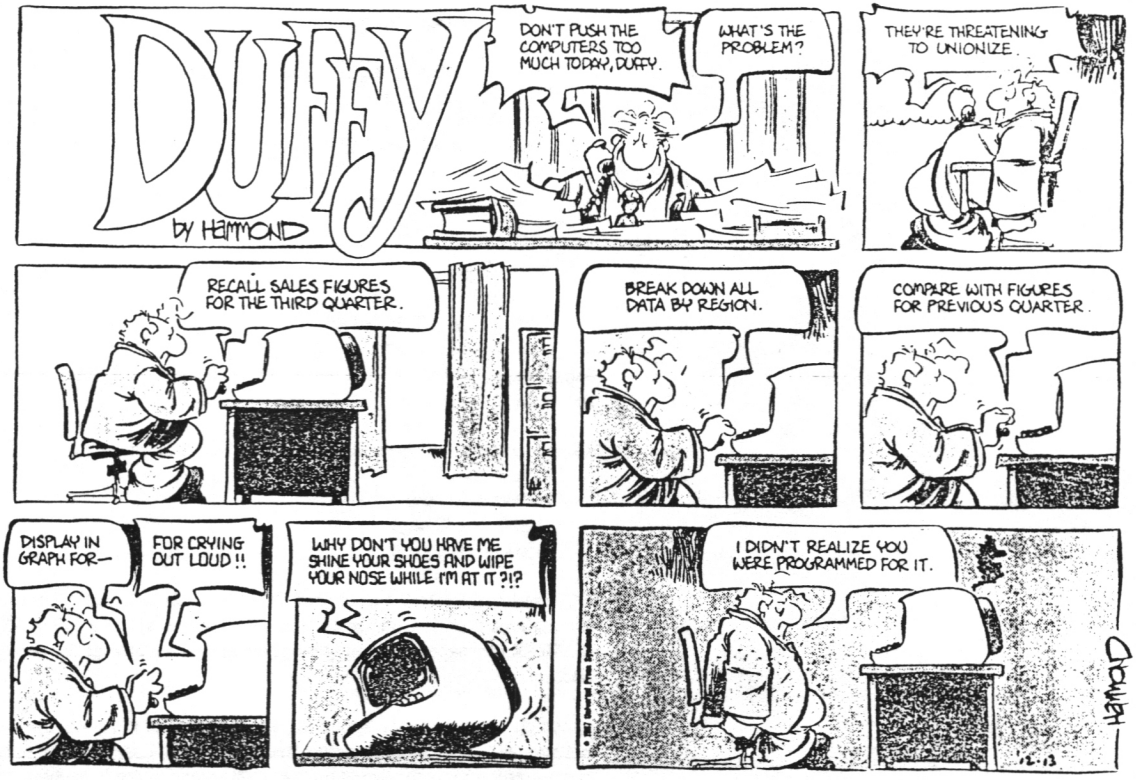
****

```
   5 PRINT TAB 9;"NICOMACHUS"
  10 LET A$="WHEN DIVIDED BY "
  20 LET B$=" ITS REMAINDER IS?"
  30 PRINT "THINK OF A NUMBER FR
OM 1 TO 100."
  40 PRINT A$;3;B$;
  50 INPUT A
  60 PRINT A
  70 PRINT A$;5;B$;
  80 INPUT B
  90 PRINT B
 100 PRINT A$;7;B$;
 110 INPUT C
 120 PRINT C
 130 PRINT "LET ME THINK A MOMEN
T..."
 140 LET Y=70*A+21*B+15*C
 150 IF Y<106 THEN GOTO 180
 160 LET Y=Y-105
 170 GOTO 150
 180 PRINT "YOUR NUMBER WAS ";Y;
", RIGHT?(Y/N)"
 190 INPUT Q$
 210 IF CODE Q$=62 THEN PRINT "H
OW ABOUT THAT."
 220 IF CODE Q$=51 THEN PRINT "I
 THINK YOU MISCALCULATED."
 230 PRINT "WANT TO TRY ANOTHER?
(Y/N)"
 240 INPUT D$
 245 CLS
 250 IF CODE D$=62 THEN RUN
 260 IF CODE D$=51 THEN PRINT "O
KAY, THANK YOU, GOODBYE."
```

```
  1 REM "ODDONEOUT"
        2K T/S1000

  2 FAST
 10 CLS
 20 PRINT "SPOT THE ODD ONE OUT
:";;;;;"
 30 FOR J=1 TO 5
 40 PRINT "
 50 NEXT J
 60 PRINT "
100 FOR L=4 TO 8
110 FOR C=1 TO 5
120 LET Z=INT (RND*20)+1
130 LET Z=Z+118*(Z>10)
131 IF Z=136 THEN GOTO 120
140 PRINT AT L,C;CHR$ Z;TAB C+8
;CHR$ Z;TAB C+16;CHR$ Z;TAB C+24
;CHR$ Z
150 NEXT C
160 NEXT L
170 PRINT AT 11,3;1;TAB 11;2;TA
B 19;3;TAB 27;4
180 LET N=INT (RND*4)
190 LET L=INT (RND*5)+4
200 LET C=INT (RND*5+1)+N*8
210 PRINT AT L,C;
220 LET Z=INT (RND*20)+1
230 LET Z=Z+118*(Z>10)
231 IF Z=136 THEN GOTO 220
240 IF Z=PEEK (PEEK 16398+256*P
EEK 16399) THEN GOTO 220
250 PRINT CHR$ Z
260 LET N=N+1
270 SLOW
271 LET J=CODE INKEY$-28
275 IF J<1 OR J>4 THEN GOTO 271
280 IF J<>N THEN GOTO 330
290 PRINT AT 15,0;"CORRECT"
295 SLOW
300 GOTO 340
330 PRINT AT 15,0;"WRONG"
340 SLOW
345 IF INKEY$>"" THEN GOTO 345
350 PRINT AT L,C;"X";AT L,C;CHR
$ Z
355 PRINT AT 18,0;"PRESS S TO S
TART AGAIN"
360 IF INKEY$="" THEN GOTO 350
370 FAST
375 IF INKEY$="S" THEN RUN
380 RUN 10
```

****



VSUG

The Vancouver Sinclair Users Group has been in existence since 1982. We are a support group for the owners and users of all SINCLAIR and TIMEX computers.

Pres:- Gerd Breunung PH*(604) 931-5509
V/Pres:- Glenn Read
Sec:- Harvey Taylor
Treas. & N/L Editor:- Rod Humphreys

Our membership dues are only $15.00/year and may be sent to the Treasurer:

Rod Humphreys
2006 Highview Place
Port Moody, B.C., V3H 1N5

Members of VSUG receive a monthly issue of ZXAppeal - our newsletter.

ZXAppeal accepts advertising. Our **PREPAID** rates are:

$10.00 -- full page
$8.00 -- 1/2 page
$5.00 -- 1/4 page

ZXAppeal is distributed to approx 30 other SINCLAIR User Groups throughout North America as well as overseas via the NETWORK. NETWORK correspondence may be directed to the Editor at the above address.